



Red Hat Ansible Inside 1.1

Red Hat Ansible Inside レポートガイド

Red Hat Inside のレポートについて

Red Hat Inside のレポートについて

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

レポートを使用して、Red Hat Ansible Inside がどのように使用されているかについて洞察を深めます。

目次

多様性を受け入れるオープンソースの強化	3
第1章 RED HAT ANSIBLE INSIDE のレポート	4
1.1. メトリクスのレポートについて	4
1.2. メトリクスファイルについて	4
第2章 メトリクスファイルの場所	6
2.1. メトリクスファイルのデフォルトの場所	6
2.2. メトリクスの保存場所のカスタマイズ	6
2.3. メトリクスファイルの表示	6
第3章 RED HAT へのデータのレポート	8

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 RED HAT ANSIBLE INSIDE のレポート

Red Hat Ansible Inside のサブスクリプション要件に準拠するには、自動化の使用状況に関するメトリクスファイルを Red Hat に送信する必要があります。このガイドでは、収集されるデータ、その収集の実装方法、データを Red Hat に返す方法について説明します。

1.1. メトリクスのレポートについて

Ansible Inside は、自動化ジョブに関する使用状況概要データを保存します。Red Hat はこのデータを使用して、製品の改善と新機能のどこに注力すべきかを特定します。

自動化ジョブが完了すると、メトリクスが収集され、ジョブに関する次の情報が保存されます。

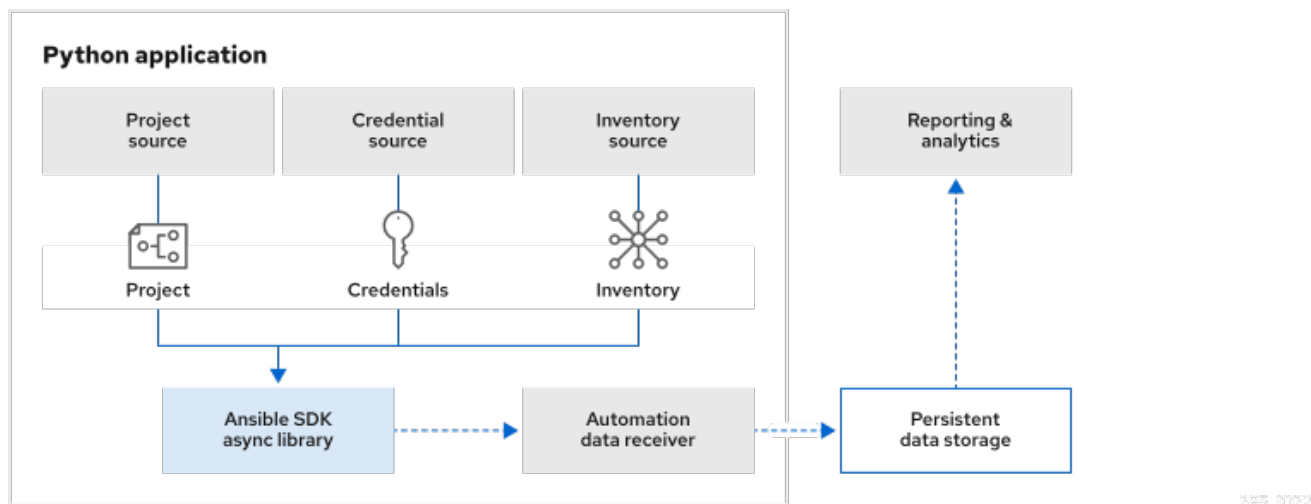
- 自動化ジョブのタイプ、期間、および時間。
- ジョブで使用されるコレクションとロールの名前、およびそれらが使用された回数。
- インストール、更新、失敗、およびスキップされたノードの数。
- ジョブ中に発生したイベント。



注記

Red Hat は、IP アドレス、場所、ユーザーの詳細、またはオペレーティングシステムの仕様などの個人情報 (PII) を収集しません。

次のアーキテクチャー図は、Red Hat に送信できるように、Python アプリケーションの外部にある永続データストレージにデータを保存する方法を示しています。



1.2. メトリクスファイルについて

メトリクスデータは、tarball ファイルにバンドルされています。ジョブごとに個別のファイルが生成されます。ファイルは暗号化されておらず、ファイルの内容は表示できます。

各 tarball ファイルには、次のような暗号化なしの CSV ファイルが含まれています。

- **jobs.csv** には、自動化ジョブの期間とステータス、実行されたタスクの数、およびジョブの影響を受けたホストの数が記録されます。

- **modules.csv** には、モジュール名、タスク数、および期間が記録されます。
- **collections.csv** には、コレクション名、タスク数、および期間が記録されます。
- **roles.csv** には、ロール名、タスク数、および期間が記録されます。
- **playbook_on_stats.csv** には、イベント ID とデータが記録されます。

第2章 メトリクスファイルの場所

Red Hat へのメトリクスのレポートは必須です。Ansible SDK をインストールすると、自動化ジョブのログメトリクスが自動的に有効になります。無効にすることはできません。

自動化ジョブが実行されるたびに、新しい tarball が作成されます。ユーザーは、ストレージの場所からデータを消去し、ディレクトリーのサイズを監視する責任があります。

Playbook を実行する各 Python ファイルのメトリクスの保存場所をカスタマイズすることも、デフォルトの場所を使用することもできます。

2.1. メトリクスファイルのデフォルトの場所

Ansible SDK をインストールすると、デフォルトのメトリクスの保存場所が `~/.ansible/metrics` ディレクトリーに設定されます。

自動化ジョブが完了すると、メトリクスがディレクトリー内の tarball に書き込まれます。ディレクトリーが存在しない場合、Ansible SDK はディレクトリーを作成します。

2.2. メトリクスの保存場所のカスタマイズ

Playbook を実行する Python ファイルに、メトリクスファイルの保存ディレクトリーへのパスを指定できます。

Python 自動化ジョブファイルごとに異なるディレクトリーパスを設定することも、複数のジョブの tarball を1つのディレクトリーに格納することもできます。Python ファイルでパスを設定しない場合、実行するジョブの tarball はデフォルトディレクトリー (`~/.ansible/metrics`) に保存されます。

手順

1. メトリクスデータを保存するファイルシステム上の場所を決定します。場所が読み取りおよび書き込み可能であることを確認してください。ディレクトリーが存在しない場合、Ansible SDK はディレクトリーを作成します。
2. Python ファイルの `main()` 関数の `job_options` で、`metrics_output_path` パラメーターを、tarball を保存するディレクトリーに設定します。
次の例では、`pb.yml` Playbook が実行された後、メトリクスファイルが `/tmp/metrics` ディレクトリーに保存されます。

```
async def main():
    executor = AnsibleSubprocessJobExecutor()
    executor_options = AnsibleSubprocessJobOptions()
    job_options = {
        'playbook': 'pb.yml',
        # Change the default job-related data path
        'metrics_output_path': '/tmp/metrics',
    }
```

2.3. メトリクスファイルの表示

自動化ジョブが完了したら、データを保存するために指定したディレクトリーに移動し、ファイルを一覧表示します。

新しく完了したジョブのデータは、名前が自動化ジョブの実行日時で始まる tarball ファイルに含まれています。たとえば、次のファイルは、2023年3月8日の午前2時30分に実行された自動化ジョブのデータを記録します。

```
$ ls
```

```
2023_03_08_02_30_24__aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaaa_job_data.tar.gz
```

tarball 内のファイルを表示するには、**tar xvf** を実行します。

```
$ tar xvf 2023_03_08_02_30_24__aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaaa_job_data.tar.gz
```

```
x jobs.csv
x modules.csv
x collections.csv
x roles.csv
x playbook_on_stats.csv
```

次の例は、**jobs.csv** ファイルを示しています。

```
$ cat jobs.csv
```

```
job_id,job_type,started,finished,job_state,hosts_ok,hosts_changed,hosts_skipped,hosts_failed,hosts_unreachable,task_count,task_duration
84896567-a586-4215-a914-7503010ef281,local,2023-03-08 02:30:22.440045,2023-03-08 02:30:24.316458,,5,0,0,0,0,2,0:00:01.876413
```

パラメーターの値が使用できない場合、CSV ファイルの対応するエントリーが空になります。上記の **jobs.csv** ファイルでは、**job_state** 値は使用できません。

第3章 RED HAT へのデータのレポート

サブスクリプション契約では、会計処理用に **.tar** ファイルを Red Hat に送信する必要があります。Red Hat パートナーの担当者が、Red Hat にファイルを送信する方法について説明します。